Bootstrapping incremental dialogue systems from minimal data: the generalisation power of dialogue grammars

Arash Eshghi, Igor Shalyminov, Oliver Lemon
Heriot-Watt University

Presenter: Prashant Jayannavar
Problem

- Inducing task-based dialog systems
  - Example: Restaurant search
Motivation

- Poor data efficiency

- Annotation costs
  - task-specific semantic/pragmatic annotations

- Lack of support for natural spontaneous dialog/incremental dialog phenomena
  - E.g.: “I would like an LG laptop sorry uhm phone”, “we will be uhm eight”
Contributions

- Solution
  - An incremental semantic parser + generator trained with RL
  - End-to-end method

- Show the following empirically:
  - Generalization power
  - Data efficiency
Background

- DS-TTR parsing (Dynamic Syntax - Type Theory with Records)
  - Dynamic Syntax
    - word-by-word incremental and semantic grammar formalism

- Type Theory with Records
  - Record Types (RTs): richer semantic representations
Background

- DS-TTR parsing (Dynamic Syntax - Type Theory with Records)

\[
\begin{align*}
\left[ \begin{array}{c}
event \\
p_1 = \text{today(event)} \\
p_2 = \text{pres(event)} \\
x = \text{robin} \\
p_3 = \text{subj(event,x)} \\
p_3' = \text{from(event,x)}
\end{array} \right] & \implies \\
\left[ \begin{array}{c}
event = \text{arrive} \\
p_1 = \text{today(event)} \\
p_2 = \text{pres(event)} \\
x = \text{robin} \\
p_3 = \text{subj(event,x)} \\
p_3' = \text{from(event,x)}
\end{array} \right] & \implies \\
\left[ \begin{array}{c}
event = \text{arrive} \\
p_1 = \text{today(event)} \\
p_2 = \text{pres(event)} \\
x = \text{robin} \\
p_3 = \text{subj(event,x)} \\
p_3' = \text{from(event,x)}
\end{array} \right]
\end{align*}
\]

"A: Today" \implies "..Robin arrives" \implies "B: from?" \implies "A: Sweden"
- Treat natural language generation (NLG) and dialog management (DM) as a joint decision problem
  - Given a “dialog state” decide what to say

- Learn to do this through learning a policy ($\pi : S \rightarrow A$) -- RL

- Define “dialog state” using output of the DS-TTR parser
BABBLE

- Inputs:
  - A DS-TTR parser
  - A dataset $D$ of dialogs in target domain

- Output:
  - Policy $\pi : S \rightarrow A$ (given a “dialog state” deciding what to say)
BABBLE

- MDP setup
  - S: set of all dialog states (induced from dataset D)
  - A: set of all actions (words in the DS lexicon)
  - G_d: Goal state
  - R: reaching G_d while minimizing dialog length
BABBLE

- Dialog state:
  - Between SYSTEM and USER utterances and between every word of SYSTEM utterances
BABBLE

- Dialog state:
  - Between SYSTEM and USER utterances and between every word of SYSTEM utterances

SYSTEM: \([S_0]\) What \([S_1]\) would \([S_2]\) you \([S_3]\) like \([S_4]\) ? \([S_5 = S_{\text{trig}_1}]\)

USER: A phone \([S_6]\)

SYSTEM: by \([S_7]\) which \([S_8]\) brand \([S_9]\) ? \([S_{10} = S_{\text{trig}_2}]\)

USER: ...
BABBLE

- Dialog state:
  - Between SYS and USER utterances and between every word of SYS utterances
  - Context up until that point in time
  - Context $C = <c_p, c_g>$
BABBLE

- SYSTEM: What would you like?
USER: A phone
SYSTEM: by which brand? [S_10]
Sys: What would you like?
Usr: a phone

\[
\begin{align*}
x_2 & : e \\
e_2 = \text{like} & : es \\
x_1 = \text{USR} & : e \\
p_2 = \text{pres}(e_2) & : t \\
p_5 = \text{subj}(e_2, x_1) & : t \\
p_4 = \text{obj}(e_2, x_2) & : t \\
p_{11} = \text{phone}(x_2) & : t \\
\end{align*}
\]

Sys: by which brand?

\[
\begin{align*}
x_2 & : e \\
e_2 = \text{like} & : es \\
x_1 = \text{USR} & : e \\
p_2 = \text{pres}(e_2) & : t \\
p_5 = \text{subj}(e_2, x_1) & : t \\
p_4 = \text{obj}(e_2, x_2) & : t \\
p_{11} = \text{phone}(x_2) & : t \\
x_3 & : e \\
p_{10} = \text{by}(x_2, x_3) & : t \\
p_9 = \text{brand}(x_3) & : t \\
p_{10} = \text{question}(x_3) & : t \\
\end{align*}
\]
BABBLE

- Dialog state:
  - Between SYS and USER utterances and between every word of SYS utterances
  - Context up until that point in time
  - Context $C = \langle c_p, c_g \rangle$
  - State encoding function $F: C \rightarrow S$ maps context to a binary vector
Grounded Semantics

\[
\begin{align*}
  x_2 &: e \\
  e_{2=like} &: es \\
  x_{1=USR} &: e \\
  p_{2=pres(e_2)} &: t \\
  p_{5=subj(e_2,x_1)} &: t \\
  p_{4=obj(e_2,x_2)} &: t \\
  p_{11=phone(x_2)} &: t \\
\end{align*}
\]

Current Turn Semantics

\[
\begin{align*}
  x_2 &: e \\
  e_{2=like} &: es \\
  x_{1=USR} &: e \\
  p_{2=pres(e_2)} &: t \\
  p_{5=subj(e_2,x_1)} &: t \\
  p_{4=obj(e_2,x_2)} &: t \\
  x_3 &: e \\
  p_{10=by(x_2,x_3)} &: t \\
  p_{9=brand(x_3)} &: t \\
  p_{10=question(x_3)} &: t \\
\end{align*}
\]

Dialogue so far

SYS: What would you like?
USR: a phone
SYS: by which brand?

RT Feature:

\[
\begin{align*}
  x_{10} &: e \\
  p_{15=brand(x_{10})} &: t \\
  e_{3=like} &: es \\
  p_{2=pres(e_3)} &: t \\
  x_8 &: e \\
  p_{14=by(x_8,x_{10})} &: t \\
\end{align*}
\]

State:

\[
\begin{align*}
 \text{Current Turn:} & \quad 1, 1, 1, 1, 1 \\
 \text{Grounded:} & \quad 0, 1, 0, 1 \\
\end{align*}
\]
BABBLE

- Dialog state:
  - Between SYS and USER utterances and between every word of SYS utterances
  - Context up until that point in time
  - Context C = \langle c_p, c_g \rangle
  - State encoding function F: C \rightarrow S maps context to a binary vector
RL to solve the MDP


USER: A phone [S_6] <- Simulated User


USER: … <- Simulated User

SYSTEM: …
BABBLE

User simulation

- Generate user turns based on context
- Monitor system utterance word-by-word
BABBLE

User simulation

- Generate user turns based on context
  - Run parser on dataset D and extract rules of the form:
    
    \[ S_{\text{trig}_i} \rightarrow \{u_1, u_2, \ldots, u_n\} \]

    \( S_{\text{trig}_i} = \) a trigger state

    \( u_i = \) user utterance following \( S_{\text{trig}_i} \) in D
What would you like?

A phone

by which brand?
BABBLE

User simulation

- Generate user turns based on context
- **Monitor system utterance word-by-word**
  - After system generates a word, check if new state **subsumes** one of the S trillion
  - If not, penalize system and terminate learning episode
What would you like? 

A phone 

by which brand?
Evaluation

- 2 datasets to test generalization:
  - bAbI
    - Dataset of dialogs by Facebook AI Research
    - Goal oriented dialogs for restaurant search
    - API call at the end
Evaluation

- bAbI+
  - Add incremental dialog phenomena to bAbI

- Hesitations: “we will be uhm eight”
- Corrections: “I would like an LG laptop sorry uhm phone”

- These phenomena mixed in probabilistically
  - Affect 11336 utterances in the 3998 dialogs
Evaluation

- **Approach to compare to (MEMN2N):**
  - Bordes and Weston 2017: Learning end-to-end goal-oriented dialog
  - Uses memory networks
  - Retrieval based model
Evaluation

- **Experiment 1: Generalization from small data**
  - Do not use the original system for a direct comparison
    - Use a retrieval based variant

- 1-5 examples from bAbI train set
- Test on 1000 examples from bAbI test set
- Test on 1000 examples from bAbI+ test set
## Evaluation

- Experiment 1: Generalization from small data
  - Metric: Per utterance accuracy

<table>
<thead>
<tr>
<th># of training dialogues:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BABBLE on bAbI</strong></td>
<td>67.12</td>
<td>73.36</td>
<td>72.63</td>
<td>73.32</td>
<td>74.08</td>
</tr>
<tr>
<td><strong>MEMN2N on bAbI</strong></td>
<td>2.77</td>
<td>59.15</td>
<td>70.94</td>
<td>71.68</td>
<td>72.6</td>
</tr>
<tr>
<td><strong>BABBLE on bAbI+</strong></td>
<td>59.42</td>
<td>65.27</td>
<td>63.45</td>
<td>64.34</td>
<td>65.2</td>
</tr>
<tr>
<td><strong>MEMN2N on bAbI+</strong></td>
<td>0.22</td>
<td>56.75</td>
<td>68.65</td>
<td>71.84</td>
<td>73.2</td>
</tr>
</tbody>
</table>
Evaluation

- Experiment 2: Semantic Accuracy
  - Metric: Accuracy of API call
  - BABBLE: 100% on both bAbI and bAbI+
  - MEMN2N: Nearly 0 on both bAbI and bAbI+
  - MEMN2N (when trained on full bAbI dataset): 100% on bAbI and only 28% on bAbI+
Summary

- An incremental semantic parser + generator trained with RL

- End-to-end training

- Support incremental dialog phenomena

- Showed the following empirically:
  - Generalization power
  - Data efficiency